Learning by Reading: Extending & Localizing Against a Model

Scott Friedman[1] **Mark Burstein**[1] **David McDonald**[1] FRIEDMAN/BURSTEIN/DMCDONALD@SIFT.NET
Amandalynne Paullada[1] **Alex Plotnick**[1] APAULLADA/APLOTNICK@SIFT.NET
Rusty Bobrow[2] ROBERT.BOBROW@GMAIL.COM
Brent Cochran[3] BRENT.COCHRAN@TUFTS.EDU
James Pustejovsky[4] **Peter Anick**[4] JAMESP/PANICK@BRANDEIS.EDU
[1]SIFT, 319 1st Ave North, Suite 400, Minneapolis, MN 55401 USA
[2]Bobrow Computational Intelligence, LLC
[3]Tufts University School of Medicine, 136 Harrison Ave., Boston, MA 02111, USA
[4]Dept of Computer Science, Brandeis University, 415 South Street, Waltham, MA 02454, USA.

Abstract

This paper describes R3 (*Reading, Reasoning, and Reporting*), our system for deep language understanding and extension of a mechanistic model for biochemical signaling pathways. The overall purpose of R3 is to read and incorporate into its model information about signaling pathways from PubMed Central journal articles. Its initial background model of these biochemical pathways is derived from an imported curated model of biological events, complexes, and proteins (reactome.org). We describe some significant issues for deep semantic parsing in this domain and how we use pre- and post-analysis reasoning to bridge the differences between the semantic information that can be derived from a text and the codified mechanistic information in the curated biomedical database. We also present extensions to relational structure-mapping to detect corroboration between the semantic parse and the model and extend the model with analogical inferences from the parse. We close with a description of empirical results with R3, including semantic parsing, model extension, grounding entity and event references, and modeling entity behavior using knowledge learned by reading.

1. Introduction

Machine reading does not end with a parse or even with a semantic interpretation of text. When we read to inform ourselves, we use our current model of the world to guide our interpretation of the text, and then we reconcile this interpretation with our model to determine consistency with our prior beliefs and perhaps to accept and incorporate the new information. Our interpretation might corroborate, extend, or conflict with our prior model and perhaps cause us to revise or extend it. We refer to this model-centric activity as *reading with a model*. Reading with a model is the central goal of our ongoing work on the *Reading, Reasoning, and Reporting* (R3) cognitive system, as part of DARPA's Big Mechanism program (Cohen, 2015).[1] R3 reads articles in molecular biology to extend and revise its models of biological mechanisms, specifically those having to do with signaling pathways.

A central capability— and research challenge— for cognitive systems that read with a model is *localizing* (i.e., recognizing and retrieving) entities and events mentioned in the text when they appear in the prior model, in order to begin the process of reconciliation. Localization allows the system to establish a mapping between the interpreted text and the model to enable bidirectional information flow between the model and the text interpretation process. That is, we seek to transfer information about mechanisms gleaned from the text into

1. This work was supported by Contract W911NF-14-C-0109 with the US Defense Advanced Research Projects Agency (DARPA) and U.S. Army Research Office. Approved for Public Release, Distribution Unlimited. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

the model (**interpretation-to-model**), either to extend the model or to identify and annotate conflicts. If localization first establishes correspondence between parts of the model and the text, we can also improve the text interpretation process (**model-to-interpretation**) by making the reading system aware of details about known entities and processes (such as their types and relations to other mentioned entities) so that when they are mentioned only by reference, those references are not overly vague or ambiguous. If known entities and events in the text are not localized correctly within the model, then the interpretation is less successful since new related information is not properly integrated.

Building a system that reads and localizes to a pre-existing biological pathway model has been developed and curated by domain experts involves many domain-general and domain-specific challenges, among them:

- Texts frequently use the same word to mean different types of objects in the model (e.g., "*RAS*") can refer to a protein, a gene, or a larger multi-protein complex, within a single article.
- Texts may describe things at different levels of abstraction than the model. For example, authors frequently talk about the the *function* of events while a purely mechanistic model may only describe the biochemical reactions taking place.
- One process or event may be part of many other processes or events in the domain model.

Our initial attempts at localization consisted of parsing articles and attempting to directly match semantic descriptions to known events and entities in a pathway model extracted from the Reactome (reactome.org) portion of Pathway Commons, a large network of reactions represented using a small OWL ontology. We ran into all of these issues in the process. There were several key kinds of mismatches between the semantic representations from parsed texts and the chemical reaction ontology (BioPAX) of the model. Some of these differences had to do with the explicit terms for kinds of reactions/reaction products.

A commonly mentioned class of named reactions are the *post-translational modifications*, including *phosphorylation* (binding a phosphoryl group to a molecule). Another is the formation of complexes with multiples of the same molecule. *Dimerization* is the binding of two like molecules to form a *dimer*. These sorts of named events were present in the model but only by implication. In the formally represented BioPAX model one has to compare the reactants and products to detect these types of events. For structure mapping to be effective, inferences explicitly identifying these reaction types are needed to make the mapping process effective. We discuss those inferences required for localization in Section 3.1.

It also became clear that articles often talk about processes at a *functional level*, in terms of triggers for and preventers of events or event sequences. Typically, in the signaling domain, processes are "switched" on or off, and proteins can "activate" or "inhibit" processes. Proteins are described as in an "activated" state when they are bound to other molecules in ways that enable them to act as catalysts in subsequent reactions.

This kind of association of functional states with molecules is so common that, in many cases, localization by matching is not effective unless the underlying model represents these molecular states explicitly. What makes a protein is "active" can be many different things at the structural, molecular level. For example, proteins like MEK and ERK are activated when they are phosphorylated. Others are *de*activated when they are phosphorylated. Some are activated when they are dimerized, etc.

To capture the specific states of activation for the particular proteins in our model, we needed a source of information about the complexes involving those proteins when they were considered "active" or "inactive". We found our source by parsing the textual summary statements associated with each reaction in the model by the original curators. For example:

- "SOS1 is the guanine nucleotide exchange factor (GEF) for RAS. SOS1 *activates* RAS nucleotide exchange *from the inactive form* (bound to GDP) *to an active form* (bound to GTP)."
- "EGFR phosphorylates PLC-gamma1, thus *activating* it."
- "*Activated MAPK proteins* negatively regulate MAP2K1:MAP2K2 heterodimers..."

These examples of reaction summaries show how the language used for human consumption conveys the functional states of their primary participants, rather than their chemical changes. R3 learns what it needs for localization by structure mapping comparisons of those descriptions against the associated model of the chemical reactions. When subsequent texts also describe functional states, we can now identify the corresponding states in our model.

R3 integrates deep semantic parsing, ontology mapping, interpretation-to-model structure-mapping, and functional reasoning. Deep parsing (Section 3.2) allows R3 to extract precise semantics and determine entity types from local lexical context. R3's ontology mapping (Section 3.3) allows it to transfer its semantic interpretation into other ontologies to identify any and all corroborating events and entities. R3 extends structure-mapping methods (Section 3.4) to support wide-scale event recognition and retrieval. Finally, R3's mechanism-level reasoning (Section 3.7) allows it to reason about *functional* factors— such as what it means when an article describes an entity as *active*— despite lacking direct functional knowledge in its ontologies.

Section 2 outlines the problem of extracting and recognizing biological events and interactions from text, focusing on challenges for natural language understanding. Section 3 describes the R3 approach to meeting these challenges. Section 4 describes empirical evidence of our claims that (1) R3 efficiently and robustly processes large bodies of text, (2) R3's learning-by-reading improves its precision and recall of *subsequent* learning-by-reading, and (3) R3 can use information learned by reading to answer new types of queries that were not supported by its initial model. We close with a discussion of future work for R3.

2. Machine Reading in the Biology Domain

Biomedical research articles are written to be read by other professional biologists who are presumed to have the requisite technical background. The brief mention of a well-known mechanism ("*RAS/RAF/MEK/ERK Pathway*") is sufficient to evoke all of the details of the mechanism in the mind of the reader. This lets them effortlessly fill in information gaps that cannot be supplied by standard discourse techniques ("*activated upon GTP loading and deactivated upon hydrolysis of GTP to GDP*" — loaded onto or hydrolyzed from what?). We need to have knowledge sources that enable our systems to do this too.

Like other authors, biologists are under pressure to keep their articles within length limits. This leads to frequent use of compaction techniques such as describing events using nominalized verbs and packing information into them as prenominal modifiers, *e.g.*, "*EGFR and ERBB3 tyrosine phosphorylation*," "*mitogen-induced signal transduction*." This changes the usual grammatical cues (such as one would use on newswire text) and requires knowledge-rich analysis techniques if parses are to be accurate.

A further property of biomedical text is that logically related information is usually distributed across multiple sentences. The following example is typical. The classification of the sites are given in the first sentence and their identity in the second. "*We observed two conserved putative MAPK phosphorylation sites in ASPP1 and ASPP2. The ASPP1 sites are at residues 671 and 746, and the ASPP2 sites are at residues 698 and 827.*" In R3 we have enhanced our discourse history to let us combine information from both sentences into a single, logically complete, representation that specifies the binding sites on ASPP1 and ASPP2 where MAPK catalyzes phosphorylation.

3. Approach

Here we describe R3's architecture and information flow. We use Figure 1 to guide our discussion, stepping through the information flow chronologically. We begin by describing the setup and operation of the domain model and the semantic parser, and then we discuss the post-parse reasoning mechanisms and operations on the domain model.

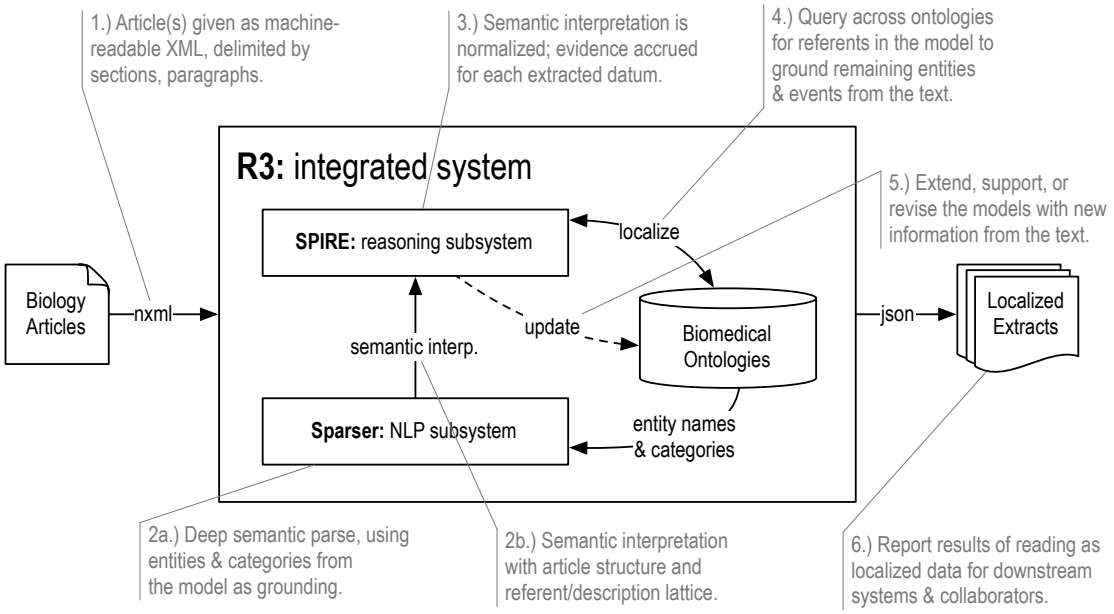


Figure 1. The R3 architecture, and the flow of information by which R3 reads articles, updates its mechanism models, and publishes extracted knowledge for human and machine collaborators.

3.1 Bootstrapping the Domain Model and the Parser

Before reading articles, R3 initializes its parser with domain vocabulary and grammar and uses inference rules to optimize and index its domain model. R3 uses the UniProt knowledge base citepuniprot2008universal as a source of protein synonyms to enhance protein recognition during parsing. It maps each protein synonym to a unique identifier to enable cross-indexing in various biological ontologies.

R3 imports OWL domain models specified in Biological Pathway Exchange (BioPAX) (Demir et al., 2010). BioPAX specifies structural information about biochemical reactions (*e.g.*, bindings, phosphorylations, and other interactions), complexes, proteins, catalysis, and reaction regulation. R3 uses domain-specific inference rules to extend the BioPAX domain model with additional structure to explicitly represent causal relations, transport events, post-translational modifications (*e.g.*, phosphorylation), functional information (activation, deactivation), and some key structural molecular categories (*e.g.*, homodimer, heterodimer) that are frequently referred to in text. Much of the enhanced content is *implicitly* described in BioPAX (*e.g.*, a homodimer is identifiable as a complex with two instances of the same protein), but R3 detects and explicitly represents this sort of additional structure to facilitate its search and localization during reading.

Finally, R3 uses a graph grammar to segment and index the enhanced BioPAX model into logical contexts. The grammar is equivalent to regular expressions over relational knowledge to describe how to traverse the model and segment it into indexable parts, *e.g.*, by starting with `biochemical-reaction` entities and traversing `left` and `right` relations to the reactions' input and output molecules, respectively, and then descending recursively through sub-molecular structures via `compound` relations, etc. This performs a regex-like match on the graph to identify matching subgraphs, and then indexes each subgraph into its own logical context. This quickly indexes the enhanced BioPAX model into smaller contexts so that R3 can quickly search the model to localize the information it reads, as we describe in Section 3.5.

3.2 Deep Semantic Parsing

The purpose of language analysis in R3 is to identify and represent the semantic content of biomedical texts to facilitate extension of a domain model and to provide a standard view of an article's content for downstream reasoners (*e.g.*, Danos et al., 2009). This requires normalizing all of the syntactic and lexical variation in how a relation is expressed to a single canonical form. References to entities and relations are also aligned with articles' document structure to facilitate search and context driven inferences. Ultimately, we seek to utilize the additional specificity of content localized from introductory remarks in papers to help the interpretation process, though this remains a goal for the future.

To do this, R3 uses the SPARSER natural language analysis platform to read the texts. SPARSER is a rule-based, type-driven semantic parser. Rules succeed only if the types of the constituents to be composed satisfy the type constraints (value restrictions) specified by the rule. SPARSER is also model driven. As described in McDonald (1996), writing a semantic grammar starts with a semantic model of the information to be analyzed along with a specification of all the ways each of the concepts can be realized in the language of the genre (*e.g.*, biomedical research articles). A compiler takes the model and creates a semantic grammar from the realization specifications by drawing on a schematic standard English syntactic grammar. This ensures that every rule in the generated grammar has an interpretation, and thus everything SPARSER is able to model it can also parse.

We use SPARSER as the parser within R3 in part due to its ability to parse into a referential model, instead of solely parsing to logical forms. SPARSER's semantic interpretations are represented in a typed lambda calculus (McDonald, 2000). The categories (predicates) are taken from an ontology whose upper structure uses Pustejovsky's model of events (Pustejovsky, 1991). There is a middle level with ontological models for location, time, people, measurement, change in amount, and more. This core is extended with a ontology of biomedical phenomena that is deliberately designed to be close to how these phenomena are described in articles in order to simplify the parsing process.

Individuals (i.e., instances of categories) represent the entities, events, and relationships that are identified when a text is read. Individuals are unique: the parsing process guarantees that every individual with a particular set of values for its properties is represented by a single object (Maida & Shapiro, 1982, McDonald, 2000). This guarantee is managed by a description lattice that tracks the addition of properties (i.e., binding of role variables). Every $\langle$*property-assignment*, *category*$\rangle$ instance is represented by a unique individual that is maintained and updated incrementally as a text is read.

Categories act as frames in a conventional knowledge representation, with a specialization lattice that permits the inheritance of realization options as well as variables (possible relations) and methods for type-specific reasoning. They are also where we state facts about normally expected properties. For example, phosphorylation events entail an active protein or other agent, a substrate protein that is phosphorylated, and a site (residue) where the phosphate is added. A residue is identified by its amino acid and its location on a particular protein. If we read about the sites of a phosphorylation and the requisite information is not supplied locally in the text, then we can assume that it is very likely to have been supplied elsewhere in the article, which motivates a search to identify it.

Our discourse component resolves pronominal and definite references using a structured history of entity and event mentions. This same facility organizes searches to expand partial descriptions of entities into full ones (frame completion) and in general to link individuals as they appear in different parts of an article. Consider this text. It compares what happens when a particular drug is or is not used:

"*In untreated cells, EGFR is phosphorylated at T669 by MEK/ERK, which inhibits activation of EGFR and ERBB3. In the presence of AZD6244, ERK is inhibited and T669 phosphorylation is blocked, increasing EGFR and ERBB3 tyrosine phosphorylation and up-regulating downstream signaling.*"

There are two mentions of the phosphorylation of residue T669 in this text, one in each sentence. The mention in the second sentence ("*T669 phosphorylation*") is marked by the sentence post-processor as being

incomplete because it does not specify the agent or the substrate. This combination of event-type and site is a unique individual stored in the description lattice. The discourse history records that this individual was also mentioned in the first sentence. This is enough to license R3 to trace up the structure on the first mention to identify the other properties it has, and to copy over any non-conflicting properties of the first to the second.[2]

3.3 Ontology Mapping

After producing a deep semantic parse with SPARSER, R3 must localize and learn from the recognized events and entities. This requires representing these events and entities using the ontology of the target domain models, which are presently described in BioPAX. Since SPARSER's interpretation is not BioPAX, R3 must perform *ontology mapping* to re-represent SPARSER's output using the R3 ontology.

R3 performs ontology-mapping using manually-created forward-chaining rules in its internal SPIRE reasoner (show in Figure 1, center). It runs these rules exhaustively, binding each rule's left-hand side to relational knowledge in the SPARSER interpretation, and asserting the corresponding right-hand side in the ontology of the domain model.[3] Since the article and the model may represent events at different granularity, the SPIRE ontology-mapping rules must occasionally generate new symbols to represent the mismatch in levels of description of entities and processes. For example, if R3 reads, *"X phosphorylates Y and Z,"* it must create two *separate* phosphorylation events for Y and Z, with X as the `agent` role for both, in order to localize them independently: these may or may not correspond to the same event in the model.

3.4 Enhanced Structure-Mapping

R3 uses SPIRE's *structure-mapping*— constrained graph-matching over relational representations based on Gentner's (1983) psychological theory of analogy and similarity— for two central operations in model localization:

1. **Retrieval**: Given a *probe* description extracted from text, recognize and retrieve all potentially corresponding entities and events from the model.
2. **Transfer**: Given a semantic description extracted from text and a description of an entity or event from the model, match the two descriptions and suggest the transfer of entities and relations into the model.

Using structure-mapping for machine reading is not a new idea; for instance, Learning Reader (Forbus et al., 2007) uses structure-mapping for offline rumination. By contrast, R3 utilizes structure-mapping for online localization (i.e., retrieval of model components) and transfer to extend the model.

The core structure-mapping operation involves computing one or more *mappings* between two representations. Each mapping is a maximal common subgraph (MCS) solution between the two representations, where each entity is a node, each relational assertion is a node, and each relation argument is a position-labeled edge. Following Falkenhainer et al.'s (1989) computational model, each of SPIRE's MCS mappings describe *correspondences* (i.e., tuples describing isomorphic nodes across graphs), a *score* that rates the quality of the correspondences, and *inferences* describing complements of the MCS (i.e., non-isomorphic relations and entities) that can be projected from one graph to the other. Structure-mapping inferences are not necessarily deductively sound, since they are based solely on structural similarity; however, in previous work, we have shown that these inferences can be practically used to revise beliefs and models (Burstein, 1988, Friedman et al., 2012). As we illustrate below, structure-mapping inferences are practical for extending the model while reading. Structure-mapping reduces the space of legal mappings— thus making the problem more tractable than traditional MCS optimization problem— by adding two additional constraints to the MCS problem:

2. The two eventualities differ in their existential status. The tense in the first sentence indicates that the phosphorylation occurs. In the second we are told that it is blocked.
3. SPIRE caches LHS clauses and definite (Horn) clause components, and it presently runs ontology-mapping at the sentence-level on SPARSER's output, so ontology-mapping is a rapid operation.

- **Tiered identicality**: Category nodes can only correspond to other category nodes with identical categories, and relation nodes can only correspond to relation nodes with identical predicates. Structure-mapping allows symbol arguments (e.g., referring to entities or events) to correspond to non-identical symbols.
- **Parallel connectivity**: If two relation or category nodes correspond, their arguments must correspond, in sequence. Applied globally: if two nodes correspond, so must their reachable subgraphs.

These two constraints drastically decrease the solution space, so SPIRE's greedy MCS algorithm is plausible and effective. Guaranteeing an optimal MCS solution is out of scope for R3 due to tractability: the decision problem for MCS is widely known to be NP-complete. As we demonstrate below, a greedy algorithm produces practical results for R3's model localization.

3.4.1 Structure-Mapping for Recognition

Computational models of structure-mapping have been used widely to compute analogies across domains, identify structural similarities, and transfer knowledge. However, event recognition and localization require much tighter matching: R3 should not retrieve events that are *similar* to an event described in a scientific article, and R3 should retrieve descriptions that could refer to the same events. We call this structure-mapping setting *recognition* rather than the more traditional setting of analogy.

While recognition differs from analogy in crucial ways that we mention below, structure-mapping still offers important benefits for flexible localization from reading. Specifically, structure-mapping supports *partial matches*: if the article mentions something not in the model (*e.g.*, a new reactant within a known reaction), structure-mapping will identify relevant candidates for model expansion.

Typically, texts will talk about specific proteins playing roles in various reactions or causal processes when in fact, and in the underlying model, these proteins are in various states of binding with other, unmentioned molecules in complexes. Hence, it is critical that the structure matching be able to identify these proteins within these larger complex structures as either reactants, products or catalysts when relating the extracted text semantics to the model.

Adapting structure-mapping to the recognition setting included the extensions that we outline below.

Identifier intersection. Like any graph-matching optimization algorithm, if structure-mapping can add a correspondence to its mapping, it will. This maximality bias yields higher-scoring mappings, but it can also produce erroneous results in an entity- or event-recognition setting. For instance, without additional constraints, the event *"SOS1 activates RAS"* will map nearly perfectly to the event *"MEK activates ERK"*; however, this is undesirable for coreference and recognition.

In its recognition setting, R3 computes *a priori* correspondence allowances, *e.g.*, so that a parsed individual can only correspond to a model individual if their identifiers (*e.g.*, list of name strings) intersect. This allows the parsed individual with namestrings {"SOS1," "SOS1_HUMAN," "SOS-1"} to correspond to the model entity with namestrings {"UniProt:Q07889 SOS1," "SOS1," "Son of sevenless protein homolog 1"} due to the "SOS1" intersection. This significantly increases recognition accuracy and reduces the search space for mappings.

Dependency constraints. Adding constraints on entities during mapping— such as only permitting two phosphorylation events to match if the phosphorylat*ed* entities also match— reduces erroneous mappings. The descriptions *"phosphorylated ERK"* and *"phosphorylated RAF"* describe the same property (i.e., phosphorylation modifications) but with non-intersecting object-roles. The phosphorylation properties are therefore incompatible for recognition purposes. We use a domain-general mechanism for specifying and mapping with dependencies, but R3 uses domain-specific rules for asserting these dependencies, *e.g.*, properties depend on their `object` role-filler. During the mapping process, when the `object` role-filler is selected for the mapping, the events are added to the search space.

Category and predicate subsumption. If R3 reads, *"SOS1 activates RAS nucleotide exchange"*, it will assert `(activates-process txt-SOS1-ent txt-RAS-NE-ent)` to describe this relationship between the SOS1 referent `txt-SOS1-ent` and the nucleotide exchange referent `txt-RAS-NE-ent`.[4] However, in the corresponding reaction in the model, R3 has described this relationship with greater specificity, *e.g.*, `(catalyzes-process-as-component mdl-SOS1-ent mdl-RAS-NE-ent)`, since SOS1 is a subcomponent of the catalyzing complex.

In R3's relational hierarchy, the `activates-process` relation from the text is a superordinate relation of the `catalyzes-process-as-component` relation in the model. SPIRE's structure-mapping algorithm supports nonidentical relation matches and nonidentical category matches albeit at a diminished score, based on the Jaccard index of their *superordinate locales*, which we define as the set of superordinate predicates or relations reachable in an upward walk of constant length k. The Jaccard index between locales is computed as $\frac{|(A \cap B)|}{|(A \cup B)|}$, so it is 0.0 (i.e., not allowed) for nonintersecting locales, 1.0 for identical locales (i.e., identical predicates or categories), and within the interval $(0, 1)$ for nonidentical predicates with intersecting locales. For R3, we use a locale distance of $k = 3$, including the relation or category itself and all relations or categories within two upward traversals. The k value is sensitive to the depth and specificity of the ontology.

Other analogy systems (*e.g.*, Falkenhainer, 1988) match nonidentical predicates and categories as a post-process. This differs from SPIRE's inclusion of nonidentical predicate matches in the initial search for correspondences.

3.5 Retrieval and Localization

After mapping the extracted information, *e.g.*, a description of an entity or process— into BioPAX, R3 localizes it by retrieving all matching entities and processes in the model. R3 uses a two-stage similarity-based retrieval algorithm, similar to MAC/FAC (Forbus et al., 1995): given a probe (i.e., the process or entity description) and a library (i.e., a set of entity and process descriptions from the model), the first stage is an efficient feature vector dot-product between the probe and each context to filter low-similarity descriptions, and the second stage is the structure-mapping recognition algorithm described above. The result is a similarity-ranked list of a subset of the model library. R3 uses *a priori* structure-mapping constraints to ensure that the explicitly-described entities and relations are in the mapping (*e.g.*, for "MEK-directed phosphorylation of ERK," the MEK, ERK, and phosphorylation event are all required); otherwise, the mapping operation terminates with a score of zero. R3 thereby identifies and ranks portions of the domain model according to their structural similarity to the extracted knowledge. As we show in Section 4, this localization approach recognizes entities and processes with high precision and recall; however, it does not account for context and causal locality, which we revisit in our discussion of future work in Section 5.

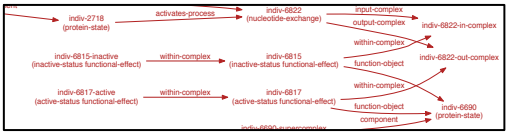
3.6 Updating the Model

After R3 interprets text (Section 3.2), maps it into BioPAX (Section 3.3), and identifies relevant portion(s) of the model (Section 3.5), it updates the model with the interpretation. The update operation is based on structure-mapping inferences (outlined in Section 3.4).

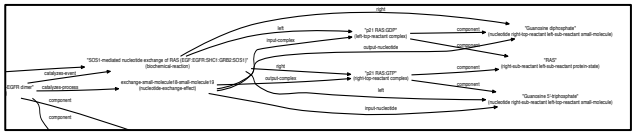
An example structure-mapping inference operation is displayed graphically in Figure 2, which illustrates the relationship between the semantic interpretation, the model, their isomorphic subgraph, and the inferred (i.e., transferred) subgraph. The semantic representation from the text (Figure 2a) and the corresponding portion of the BioPAX model (Figure 2b) are the inputs to structure-mapping, which computes the maximal common subgraph (shown in blue in Figure 2c, using the symbol names from the model). The complement (i.e., non-isomorphic portion) of the semantic interpretation provide structure-mapping inferences (shown in red in Figure 2c) that R3 transfers into the model.

4. The symbols are renamed here for clarity.

a.) Output of Semantic Parse



b.) Existing Event in Model



c.) Extended Event in Model: (**model complement, isomorphism, parse complement**)

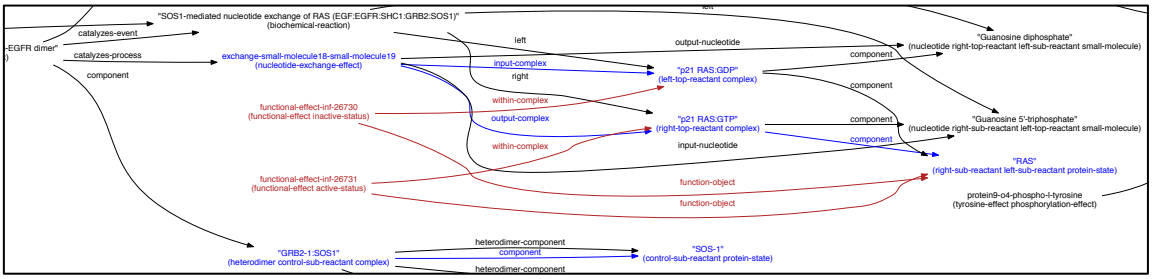


Figure 2. After parsing text, R3 maps the semantic parse into the model ontology (a, above) and uses the retrieved portion of the model (b, above) to compute a mapping (c, above). R3 uses corroborating, isomorphic structure (c, blue) as a scaffold to transfer the complement semantic structure from the parse (c, red) directly into the model to extend it.

Content-wise, the interpretation in Figure 2 corresponds to the following text:

> "SOS1 activates RAS nucleotide exchange from the inactive form (bound to GDP) to an active form (bound to GTP)."

The interpretation includes the following statements (and others) about a nucleotide exchange and a RAS protein:

```
(isa indiv-6822 nucleotide-exchange)
(input-complex indiv-6822 indiv-6822-in-complex)
(output-complex indiv-6822 indiv-6822-out-complex)
(name indiv-6690 "RAS")
```

Structure-mapping computes that the nucleotide exchange event parsed from the text is isomorphic to a known nucleotide exchange event in the model. Also, the protein, input, and output complexes of this event in the text are isomorphic to the respective protein, input, and output complexes of the event in the model. However, the semantic interpretation also contains novel (i.e., non-isomorphic) information that the RAS is *inactive* in the input complex of the nucleotide exchange and is *active* in the output complex:

```
(isa indiv-6815 inactive-status)
(function-object indiv-6815 indiv-6690)
(within-complex indiv-6815 indiv-6822-in-complex)
(isa indiv-6917 active-status)
(function-object indiv-6817 indiv-6690)
(within-complex indiv-6817 indiv-6822-out-complex)
```

The isomorphic structure (shown in blue in Figure 2c) provides a scaffold to transfer this novel information (shown in red in Figure 2c) into the model, importing new categories and relations describing existing entities and events in the model, and generating new symbols for novel events and entities.

For our evaluation described in Section 4, R3 only transfers inferences that describe protein function and behavior, such as active and inactive forms of proteins, and processes that activate and deactivate proteins. In the case of Figure 2, R3 learned that p21 RAS is active when bound to GTP and inactive when it is bound to GDP.

3.7 Mechanism-Level Reasoning and Propagation

After transferring the inferences into its model as described in the previous section, R3 propagates information throughout the model and makes secondary inferences. At present, R3 only propagates information about protein function and activation, but we are expanding the scope of propagation our ongoing work. When R3 learns that a protein instance is active or inactive, it revises all relevant reactions and super-complexes in the model, detecting changes in active status across reactions and labeling all activation or deactivation processes in the pathway. Updating the model with one fact can cause widespread revision of complexes and reactions in the model.

These updates to the model change the relational structure of the entities and reactions that R3 uses for localization, as described in Section 3.5. In Section 4, we show how R3's model extensions increase its precision when localizing subsequent interpretations from text. In this fashion, learning by reading improves R3's *subsequent* learning by reading.

4. Experiments

4.1 Evaluating Breadth & Efficiency of Information Extraction

We evaluated R3's semantic parser and its ability to extract information, filter irrelevant information (i.e., entities or events not in the domain model) and merge duplicate information against nearly 1,000 biology articles from PubMed Central. This supports our claim that R3 can efficiently and robustly perform deep semantic analysis.

We configured R3 to extract information about post-translational modifications such as phosphorylation and ubiquitination reactions, as well as positive and negative regulations of processes, and increases or decreases in molecule concentrations. Other information— including binding events, indirect causal relations, translocation events, transcription events, and more— were parsed but not analyzed with respect to the domain model in this first experiment. Additionally, R3 used epistemic filtering to ignore historical, hypothetical, or negated statements, in order to focus on positive information.

R3 read all of the articles in approximately 20 minutes. In total, it extracted 15,876 semantic descriptions of the targeted data, across all sections of all papers. This includes entities and events that were unrelated to the model, as well as duplicate data, since multiple sentences often refer to the same event.

R3 discarded 619 event descriptions that were only mentioned in the introduction or methods sections, since this experiment was focused on the new contributions of articles, and not the exposition or methodology. R3 then analyzed each extracted datum for model relevance, e.g., whether the proteins of a reaction were described in the domain model. R3 filtered out 8,864 irrelevant data, leaving 6,384. Finally, R3 merged these entities and events— and the parsed text that served as evidence— into 2,351 model-relevant data.

This test demonstrated the robustness and efficiency of R3's parsing operations. Unfortunately, since we do not have a human expert's gold standard to judge precision and recall for R3 on these 1,000 documents, this experiment does not provide evidence of R3's accuracy. That is the purpose of the next two experiments.

Here we summarize three evaluations: an information extraction evaluation (Section 4.1); a localization evaluation (Section 4.2); and a demonstration of R3 using the knowledge it learned by reading to explicate protein function and behavior (Section 4.3).

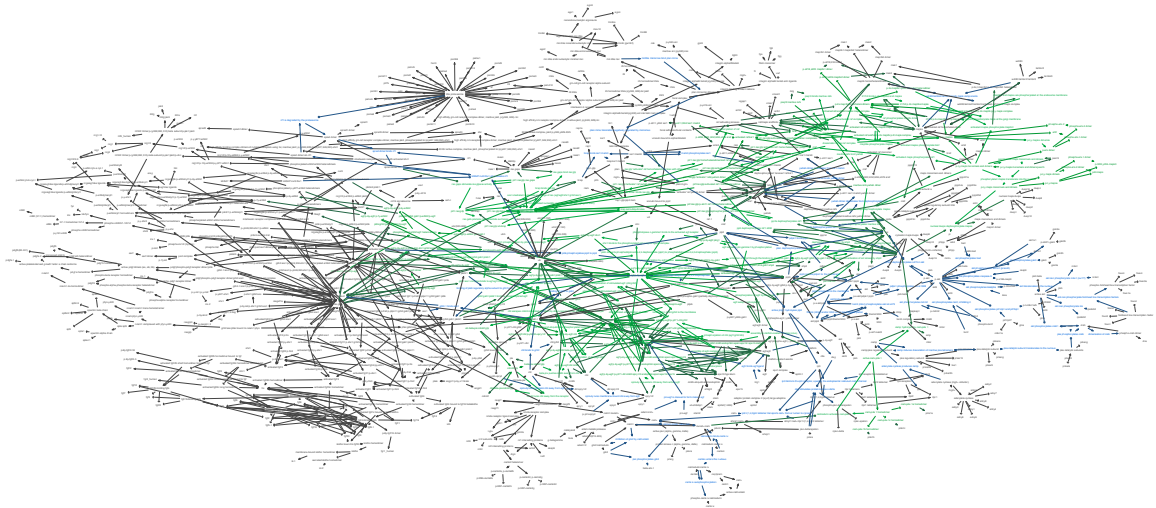


Figure 3. A graph of R3's domain model (the EGFR signaling subset of Reactome), including 911 molecule nodes and 128 biochemical reaction nodes. Each node is itself a description of the corresponding event or entity, containing an average of 110 assertions per description.

4.2 Evaluating Localization

In this experiment, we demonstrate that R3 can learn functional knowledge by reading, and that this knowledge improves R3's subsequent ability to localize extracted knowledge as it reads.

For this experiment, we used the entire "Signaling by EGFR" subset of the open-source, peer-curated Reactome pathway database.[5] Reactome pathway models describe reactions, reactants (i.e., complexes, proteins, and other molecules), catalysis and regulation relations, and protein modifications (e.g., phosphorylation, ubiquitination). The "Signaling by EGFR" Reactome subset contains 128 biochemical reactions and 911 molecules (i.e., proteins, complexes, small molecules, and other physical objects).

R3 parsed summaries (i.e., multi-sentence descriptions) and display-names (i.e., labels) of reactions that refer to molecules as "active," "inactive," "stimulated," or "activated," or alternatively that refer to "activation" or "activating" a protein. These mentions of protein activity describe functional knowledge, which the BioPAX model does not represent natively. Since the summaries and display-names are related *directly* to a corresponding reaction in the model, there is no need for R3's localization step; R3 simply maps the parser's semantic interpretation directly against the reaction in the model, and updates the model as described in Section 3.6 and Section 3.7. R3 thus reads textual passages embedded within its model to extend the model itself. We refer to this automatic process as *bootstrapping* the system with functional knowledge.

Figure 3 shows R3's bootstrapped domain model after it learns by reading summaries and display-names, where each node in the graph is a separate description of a reaction or reactant, with an average of 110 assertions each. The green nodes in R3 are portions of the model (i.e., reactions and molecules) that changed as a result of bootstrapping: reading the display-names and summaries and then propagating the information.

To qualify the effect of R3's bootstrapping on its ability to localize extracted information, we ran R3's localization operations on an article and compared the f-measure before and after. In this article, R3 extracted six mentions of biochemical processes that had correspondences in the domain model: three activations of ERK, one activation of MEK, one MEK-ERK association, and one MEK-directed phosphorylation of ERK. Before its functional knowledge from bootstrapping, R3 could *not* localize these activation mentions, but it properly localized the rest, so it scored an average F-measure of 0.33. After bootstrapping, R3 scored an

5. The BioPAX OWL files are downloadable via the pathway browser: http://www.reactome.org/PathwayBrowser/

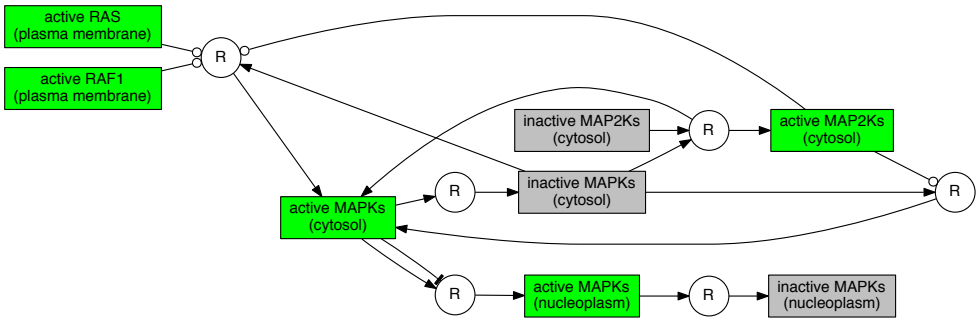


Figure 4. R3-generated graph that explains the activity (i.e., function) of Mitogen-activated protein kinase (MAPK), including the reactions that activate it and its downstream events when active.

average 0.94 (the imperfection was due to retrieving one erroneous event when localizing MAPK activation). Similarly, for localizing eight distinct molecules mentioned by the article, two of which were described as "active," R3 scored an average F1 of 0.75 without bootstrapping and 1.0 with bootstrapping.

This provides preliminary evidence that R3's learning-by-reading process of bootstrapping improves its ability to localize extracted knowledge during subsequent reading.

4.3 Demonstrating Learned Knowledge

In addition to using its learned knowledge to improve model localization, R3 can display the functional knowledge that it learned by reading. This supports the claim that after reading, R3 can answer new types of queries that were unanswerable with the initial model. Figure 4 shows a graph generated by R3 to describe the function— including activation, deactivation, and event behavior— of the MAPK protein. Before reading, R3 had no concept of "active" or "inactive" RAS/RAF/MAP2K/MAPK. After reading, R3 is able to describe the event structure of MAPK— relative to activating components RAS, RAF, and MAP2K, and including its translocation to the nucleus— which is an accurate representation of MAPK function in EGFR signaling.

In Figure 4, "R" nodes are reactions in the model, arrows from molecules to reactions indicate that the molecule is a left-hand-side (i.e., input) reactant, arrows from reactions to molecules indicate that the molecule is a right-hand-side (i.e., output) reactant, dotted arrowheads indicate that the molecule is a direct catalyst of the reaction, and tee arrowheads indicate that the molecule negatively inhibits of the reaction in a regulatory role.

This demonstrates that the information R3 learned by reading allows it to accurately reason about protein function, which is was not possible with the initial domain model R3 was given.

5. Conclusion & Future Work

We described the R3 system for reading with a model, including relevant advances in semantic parsing and structure-mapping to accurately extract information and localize it within a large third-party domain model. We described a coarse evaluation of R3's parsing capabilities, an analysis of its model localization, and we showed that R3's uses information it reads to explain protein function within a signaling pathway.

R3 presently uses semantic similarity to localize events in the model, but this is not always sufficient: context is an important consideration. Consider the sentence "*SOS and Grb2 promote the formation of GTP-bound p21 Ras*." Without more information, this will perfectly match at least 13 distinct biochemical reaction

entries in some massive BioPAX models. Distinguishing which of these perfect matches the article refers to— and it could be more than one— requires using context of the surrounding text. We are implementing a measure of *causal relevance*, so R3 can use previous, high-confidence localization operations to rank these candidates based on their proximity in the causal model. This assumes that biology articles describe causally-related events and entities rather than unrelated events and entities, which holds true in our experience.

Also, R3 presently only *extends* the model with new information, but learning by reading also involves detecting and reconciling conflicts. Important near-term future work on R3 will enable it to automatically identify possible conflicts in these extensions and then pose possible resolutions to these conflicts.

References

Burstein, M. H. (1988). Combining analogies in mental models. In *Analogical Reasoning*, (pp. 179–203). Springer.

Cohen, P. R. (2015). Darpa's big mechanism program. *Physical Biology*, *12*.

Danos, V., Feret, J., Fontana, W., Harmer, R., & Krivine, J. (2009). Rule-based modelling and model perturbation. In *Transactions on Computational Systems Biology XI*, (pp. 116–137). Springer.

Demir, E., et al. (2010). The biopax community standard for pathway data sharing. *Nature biotechnology*, *28*, 935–942.

Falkenhainer, B. (1988). *Learning from physical analogies: a study in analogy and the explanation process*. Technical report, DTIC Document.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, *41*, 1 – 63.

Forbus, K. D., Gentner, D., & Law, K. (1995). Mac/fac: A model of similarity-based retrieval. *Cognitive Science*, *19*, 141–205.

Forbus, K. D., Riesbeck, C., Birnbaum, L., Livingston, K., Sharma, A., & Ureel, L. (2007). Integrating natural language, knowledge representation and reasoning, and analogical processing to learn by reading. *Proceedings of the National Conference on Artificial Intelligence* (p. 1542).

Friedman, S. E., Barbella, D. M., & Forbus, K. D. (2012). Revising domain knowledge with cross-domain analogy. *Advances in Cognitive Systems*, *2*, 13–24.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive science*, *7*, 155–170.

Maida, A., & Shapiro, S. (1982). Intensional concepts in propositional semantic networks. *Cognitive Science*, *6*, 291–330.

McDonald, D. D. (1996). The interplay of syntactic and semantic node labels in partial parsing. In H. Bunt & M. Tomita (Eds.), *Recent Advances in Parsing Technology*, (p. 295323). Kluwer Academic Publishers.

McDonald, D. D. (2000). Issues in the representation of real texts: The design of krisp. In L. M. Iwanska & S. C. Shapiro (Eds.), *Natural Language Processing and Knowledge Representation*, (pp. 77–110). MIT Press.

Pustejovsky, J. (1991). The syntax of event structure. *Cognition*, *1*, 47–81.